RECEIVED
CENTRAL FAX CENTER
MAY 2 1 2007

REMARKS

Presently pending in the current application are claims 1-23. Claims 1, 3, 7, 9, 13, 17, and 22 are currently amended and claims 2, 4-6, 8, 10-12, 14-16, 18-21 and 23 are original.

Rejections

Drawings

The drawings were objected to because they do not include the following reference sign(s) mentioned in the description: an end 220 (Pg. 10, par. 0030, l.3), forks 260 and states 270 (Pg. 10, par. 0030, l.5). The specification has been amended to delete these reference signs. As such, Applicant respectfully believes the Figures are in condition for allowance.

Specification

The disclosure was objected to because: "a variety types" (Pg. 3), "steps in well coherent and logical to retrieve" (Pg. 14), and "allowing the end-user directly interacts with and manipulates" (Pg. 15) are improperly constructed phrases. An improper use of trademarked terms has been used such as "Java" (Pg. 6), "CORBA" (Pg. 11), "DCOM" (Pg. 11), "Internet Explorer" (Pg.12), and "Netscape" (Pg. 12). The phrases and trademarks have been corrected. As such, Applicant respectfully believes the disclosure is in proper condition.

Claims

Claim Objections: The claims were objected to because of the following informalities: "high-level structure" should be pluralized. In claims 7, 13, and 22, high-level structure" is pluralized. As such, Applicant respectfully believes the claim objections should be lifted.

Claim Rejections:

35 USC 112 - Claims 1-2, 4-8, 17-18, and 20-23 were rejected under 35 USC 112. Applicant has amended independent claims 1 and 17, as well as dependent claim 3 to correct the antecedent basis issue as well as to correct the deficiency inherent in independent claims 1 and 17. As such, Applicant respectfully believes the 35 USC 112 claim rejections should be lifted.

35 USC 103 – Claims 1-7, 9-15, and 17-22 were rejected under 35 USC 103(a) as being unpatentable over Iyengar (6,018, 627) in view of Diec (6,324, 568). Applicant does not believe that Iyengar, Diec, or any of the other cited art teach or disclose the original claims. For example, the claims of the instant invention disclose developing and executing software applications at an abstract design level. Abstract design consists of visual models that represent the application logic, yet excludes programming technologies like architecture, detailed design, programming languages, middleware, etc. (i.e. independent of the underlying programming technology). Unlike traditional development methods and practices where visual models are used for documentation purposes or to capture a subset of an application logic that requires further manual development with more detailed design models, code generation, annotation and augmentation, the claims of the instant invention describe a fully functional and semantically complete representation of the application logic (abstract design) as models that can be executed in an execution platform with no further development effort.

In contrast, Iyengar describes a development environment that facilitates the integration of various 3rd party development tools, each addressing different phase or aspect of object-oriented development, by transforming the tool specific artifacts to UML

and storing them in a shared repository. Figure 1 and the supporting description in Col. 3 l. 56 to Col. 4 l. 33 describe the development flow, which is consistent with common application development, within this environment, and uses visual models in enterprise and domain modeling (28 and 27 respectively), yet describes many further development activities like business logic development, components development/wrapping and components assembly (29, 30, and 31 respectively), before having sufficient artifacts and details to deploy the application (32) so it can be launched and executed.

More specifically, Iyengar in Figures 2a-b, 3 and 4 represent a selection mechanism (e.g. a menu) for a developer to use for selecting the appropriate tool to perform the specific phase or aspect of the development process, and by no means describes a method for *capturing application logic at an abstract level in such a manner as to be readily deployable and executable without further development.*

Col. 5, l. 21-30 explains UML and its status in the development community as the preferred notation for representing object-oriented concepts and artifacts. UML has many models (11+ models) that are useful in the different phases/aspects of object-oriented development, which goes far beyond just the abstract design, including business process definition, use case analysis, application architecture, system architecture, and detailed design. Col. 5, l. 37-59 further explains the use of UML as the unified notation for transforming tools artifacts into to be stored in a repository. These unified artifacts can be transformed into a new set of models (37-41) to become the basis for a new application. More importantly, it is used for sharing artifacts between different tools within the environment that addresses different stages of the development (42-53) and introduces a mechanism to annotate the models to accommodate tool's proprietary attributes.

Accordingly, UML may not necessarily be used by developers to capture the application logic within this development environment unless one or more of the 3rd party tools happened to use UML, which has no barring on the invention. Finally, 54-59 discusses "creating and maintaining source code for the various components" in a middleware independent form so it can be later specified in the component build stage (Col. 11, l. 26-59).

In contrast, in the preferred embodiment of the subject patent, a small subset of UML models are used, *only the ones that are concerned with abstract design*. Further UML is used as the primary method for the developer to capture application logic in the process of developing and executing a software application. ***The models are immediately executable as they get saved in the execution platform, hence, the developer can execute and validate the logic immediately with no further steps***. Finally, middleware separation (or abstraction) is a natural side effect of the development at the abstract design level, which does not involve any source code creation and maintenance all together.

Diec describes a method of web enabling a server application. It provides a system where a web server can front-end an application server to provide browser access to the application using standard HTTP requests and HTML/XML responses. It describes the use of CGI to manage communication with the application server in response to certain HTTP requests (Col. 1, l. 13-42), and Tags embedded in HTML to specify desired data from the application and the placement of this data within the HTML response (Col 4, l. 6-19). CGI's and Tags will have to be manually developed or generated, and made available to the web server for a each particular application.

In contrast, the claimed request/response in the subject invention extends the request/response of operations/processes common in object-oriented technology within an execution platform to "external client device" (i.e. entities or actors outside of the execution platform) without additional steps/effort. Accordingly, a typical operation invocation with object parameters translates to an external request with parameters in a format specific to the external client device, which then is converted by the execution platform to object parameters expected by the operation. Vice versa is the response, where the returned object(s) from the executed operation/process are converted to a format appropriate to the external client device and sent back in response to the external request (Figure 8, para. 31-36). *The claims describe this behavior in the context of executing application logic at the abstract design level (i.e. models) by simply deploying the captured models to the execution platform.*

Although Applicant does not believe that Iyengar, Diec, or any of the other cited art teach or disclose the original claims, purely in the interest of expediting the prosecution of the instant invention, Applicant has amended independent claims 1, 9, and 17 predominantly in the following manner:

Developing and executing software applications at an abstract design level, the method comprising:

capturing an application logic at the abstract design level as one or more visual models for developing a software application, the visual models being independent from an underlying programming technology;

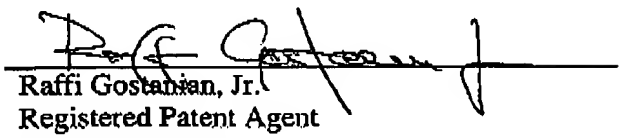
dynamically deploying the captured application logic to an execution platform, wherein the deployed application logic is immediately executable;

dynamically executing the deployed application logic from the execution platform in response to a direct external request sent by an external client device to the execution platform, the external request having one or more parameters;

processing the direct external request; returning one or more response objects after processing the direct external request; and presenting converted response objects to the external client device based on a type of the external client device or the parameters of the external request.

Neither Iyengar, Diec, or any of the other cited art teach or disclose the limitations presented in currently amended independent claims 1, 9, and 17. As such, Applicant believes currently amended independent claims 1, 9, and 17, as well as the claims that depend from them, are in condition for allowance and respectfully requests they be passed to allowance.

Respectfully Submitted,



Raffi Gostanian, Jr.
Registered Patent Agent
Reg. No. 42,595

Date: 05/21/2007

RG & Associates
1103 Twin Creeks, Ste. 120
Allen, TX 75013

972.849.1310